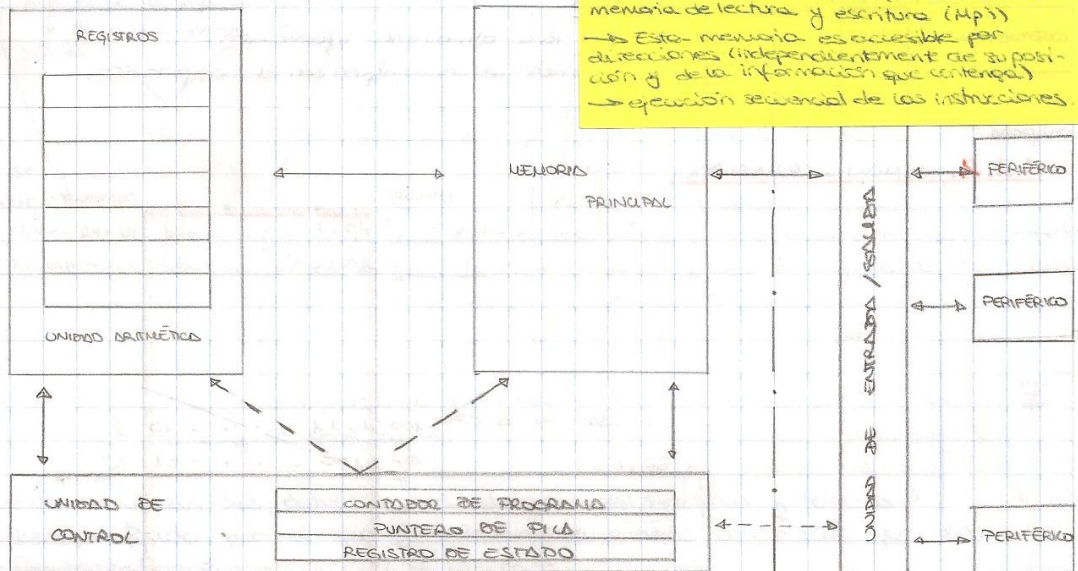


NATALIA MEDINA RIVAS

TEMA 1. INTRODUCCIÓN A LOS COMPUTADORES

1.1. ARQUITECTURA VON NEUMANN



• **INSTRUCCIONES MÁQUINA** Instrucciones u órdenes, almacenadas en la memoria principal para poder ser leídas y ejecutadas. Escritos en binario.

Los programas que ejecuta el hardware están en lenguaje máquina. El computador puede ejecutar una instrucción máquina específica. Cada computador solo puede ejecutar un conjunto de instrucciones o **juego de instrucciones** que define su arquitectura.

El modelo de computadores de Von Neumann se basa en 3 postulados:

- programa almacenado (en una única memoria de lectura y escritura (Mp))
- Esta memoria es accesible por direcciones (independientemente de su posición y de la información que contenga)
- ejecución secuencial de las instrucciones

• **MEMORIA PRINCIPAL (Mp)** compuesta por un conjunto de celdas idénticas, esto es, que tienen el mismo n° de bits (mismo tamaño). Cada celda está identificada por una dirección. Sobre la celda seleccionada se puede realizar una operación de lectura (conocer el valor almacenado previamente) o de escritura (almacenar un nuevo valor). Contienen los datos, las instrucciones y los programas.

• **LA UNIDAD ARITMÉTICA (ALU)** ^{aritmético-lógica} es la encargada de realizar las operaciones aritméticas y lógicas. Los datos sobre los que opera esta unidad provienen de la memoria principal, y pueden estar almacenados de forma temporal en algunos registros de la de la propia ALU.

• **LA UNIDAD DE CONTROL (U.C.)** se encarga de leer, una tras otra, las instrucciones máquina almacenadas en la Mp, y de generar las señales de control necesarias para que todo el computador funcione. Para conocer en todo momento la posición de memoria en la que está almacenada la instrucción que corresponde ejecutar, existe un registro llamado **contador de programa** que contiene esta información.

• **LA UNIDAD DE ENTRADA/SALIDA (U.E/S)** realiza la transferencia de información con unas unidades externas o **periféricas**, lo que permite, entre otras cosas, cargar datos y programas en la memoria principal y sacar resultados impresos.

Finalmente, conviene resaltar que existen unos **caminos** ^(flechas de trazo continuo en el dibujo) cuyo objetivo es hacer que las instrucciones y los datos circulen entre las distintas unidades del computador.

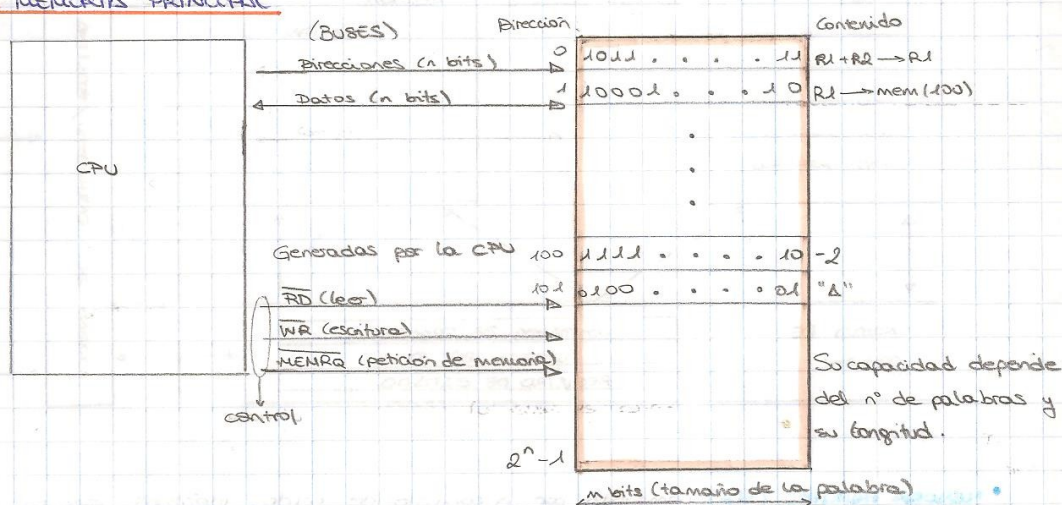
2

Se denomina **Unidad Central de Proceso (CPU)** al conjunto de la unidad de control, los registros y la ALU de un computador, es decir, el bloque que ejecuta las instrucciones. Para formar un computador hay que añadir la Mp, las unidades de E/S y las periféricas. Con frecuencia se denomina **procesador** a una CPU aunque a veces se extrapola este término al conjunto formado por una CPU y una pequeña memoria.

Las periféricas más corrientes son:

- Impresoras para producir texto
- Discos, discos compactos, etc Para almacenar información
- Terminales Para que los usuarios se comuniquen con el computador.

1.1.4. MEMORIA PRINCIPAL



La Mp contiene secuencias de bits que pueden ser instrucciones o datos o programas

Los procesadores tienen una señal desde la memoria a la CPU que le avisa cuando ha terminado el acceso.

- dirección → lectura
- dirección + datos → escritura

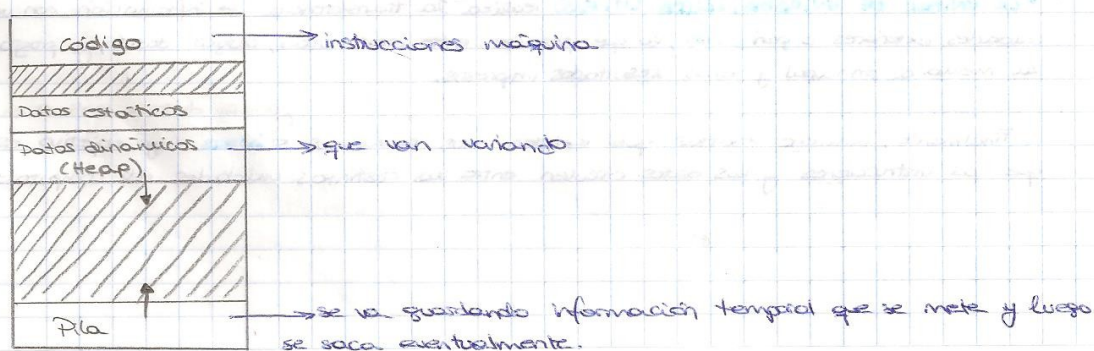
Para poder leer/escribir se activa \overline{MEMRQ} y $\overline{RD}/\overline{WR}$ según corresponda.

Para poder ~~leer~~ escribir también se activa el bus de datos y direcciones (para ambos)

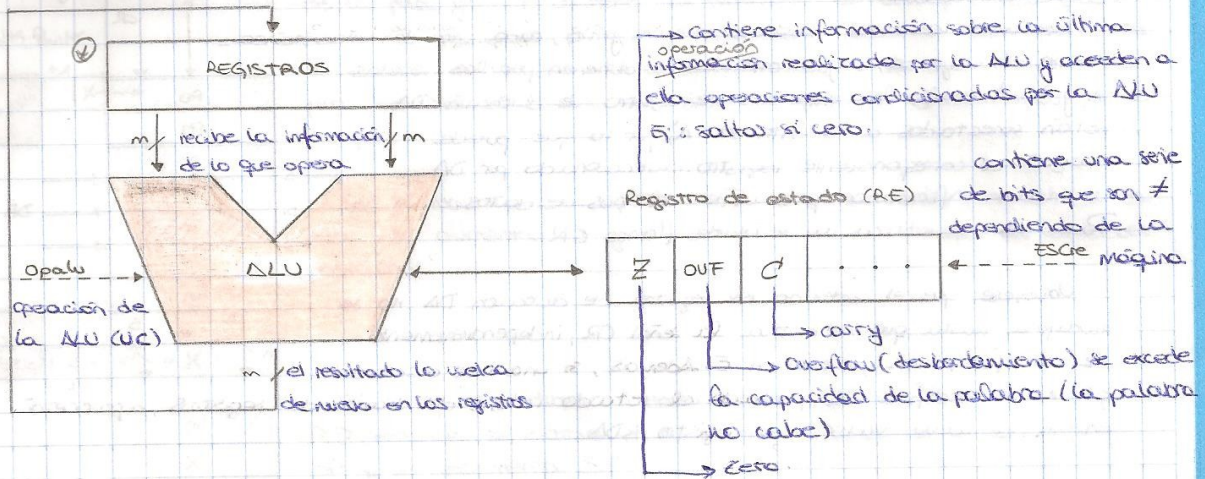
El bus de direcciones se activa tanto para lectura como escritura.

Después del tiempo de acceso, la base de datos devuelve la información a la CPU.

1.1.5. ORGANIZACIÓN DEL ESPACIO DE MEMORIA



1.1.2. UNIDAD CENTRAL DE PROCESO (CPU) / UNIDAD ARITMÉTICO-LÓGICA (ALU)



Ejemplo: si ha dado cero en la ALU se activa el bit **Z** y se pone a 1.

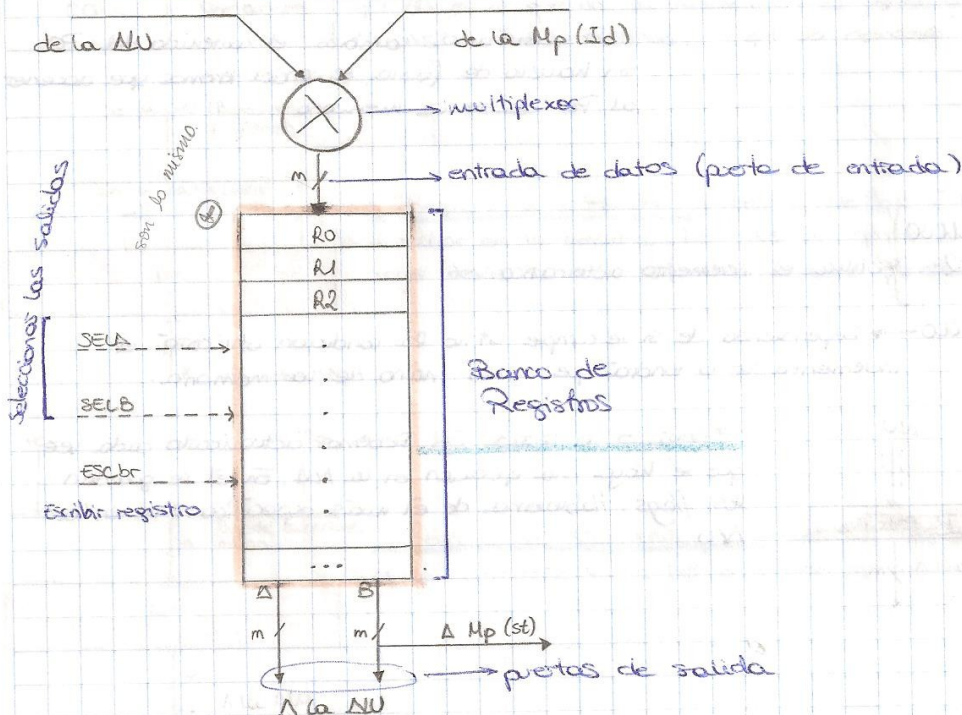
¿Dónde se ubican los datos de las instrucciones aritméticas y lógicas?

Viene definido por el modelo de ejecución del computador que ven:

- Registro - Registro (es el más utilizado)
- Registro - Memoria
- Memoria - Memoria

En estos modelos el menos restrictivo contiene a los otros.

CPU. BANCO DE REGISTROS

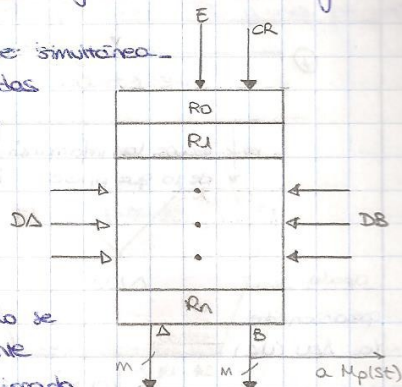


4

El Banco de Registros incluye los registros de propósito general visibles al usuario, esto es, que se pueden manejar desde el ensamblador o lenguaje máquina.

Supongamos un banco de registros, con dos puertos A y B, de salida y una E de entrada.

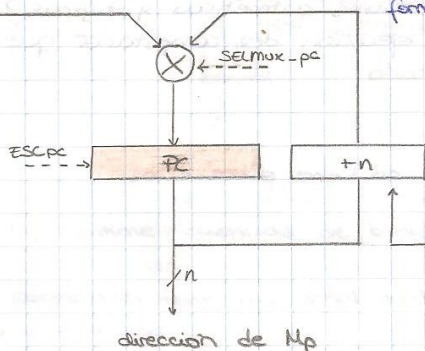
Mediante las dos direcciones DA y DB se pueden leer simultáneamente dos registros, cuyos contenidos se obtienen por las salidas A y B. Además, las entradas de registro de dirección DA están conectadas a la entrada, por lo que puede cargar el correspondiente registro direccionado por DA con nueva información proveniente del bus de datos. Para ello se utiliza la señal de flanco CR.



Obsérvese que el contenido del registro de dirección DA no se modifica hasta que se activa la señal CR, independientemente de la información presente en E. Además, si una vez accionado CR no se modifica DA, pasado el retardo interno del banco de registros, aparecerá en A el nuevo valor del registro RDA.

TIPOS DE REGISTROS de propósito general: Son los que se encuentran en este banco de registros y podemos usarlos libremente en cualquier instrucción. Que estén agrupados quiere decir que tienen señales de control comunes.

→ **de propósito específico.** Su uso está restringido a determinadas instrucciones, suele aparecer de la Mp.



Registros específicos característicos:

→ **contador de programa PC** contiene la dirección de la siguiente instrucción a ejecutar, suele trabajar de forma secuencial.

incrementador automático para avances de forma secuencial.

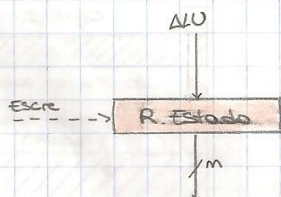
Si quisiéramos cambiar el contenido del PC sin hacerlo de forma secuencial tenemos que acceder al PC desde la memoria.

Ejemplo

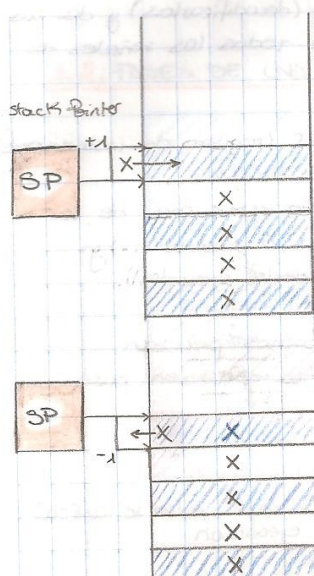
ld r1, /1000
add r1, r2
inc r1
jmp z, /2000
dec r2

utiliza el incremento automático.

Dependiendo de si se cumple o no la condición utilizará el incrementador o tendrá que echar mano de la memoria.



→ **Registro de estado SR** Debemos actualizarlo cada vez que se haga una operación en la ALU. En él se guardan los flags. Hablaremos de él más específicamente en la ALU.



→ Puntero de Pila (SP) da información sobre cuál es el último dato de la pila ya que contiene la dirección de la MIP en la que se encuentra este o la primera posición vacía. Podemos decir que apunta a la cima de la pila. En algunos procesadores no existe el puntero de pila, en ellos se utiliza un registro de propósito general como puntero de pila.

Si se realiza una operación **PUSH** para añadir un dato a la pila, se aumenta SP y se coloca el nuevo elemento en la nueva posición.

Si se quiere extraer un dato de la pila se realiza la operación **POP**, entonces se extrae el contenido del registro al que apunta SP y se decrementa SP.

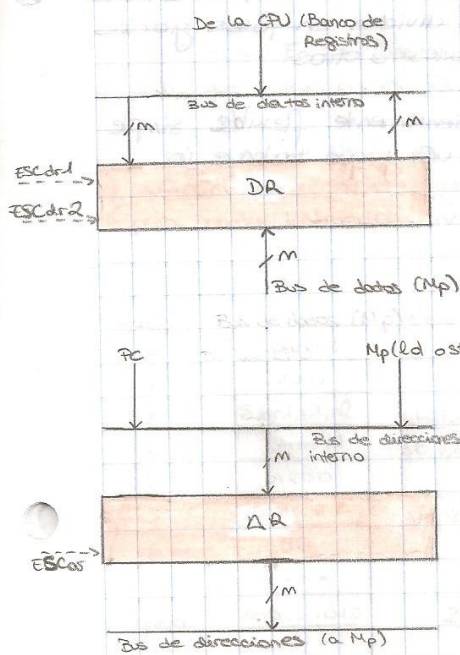
→ Registros transparentes son registros propios del procesador que, en ningún caso, pueden ser controlados o modificados por el usuario.

Registros transparentes característicos:



→ Registro de Instrucción (AI) Contiene la instrucción que se está ejecutando. Esta instrucción está organizada en partes.

Si la instrucción tiene dos palabras, en el IR solo se carga la primera (la que contiene el código de operación [CO] en cualquier caso). La segunda palabra no se sabe dónde ir, pero nunca al IR. El CO es el que da la información al procesador para activar las señales de control necesarias.



→ Registro de datos (DR) Contiene el dato que se ha leído o se va a escribir en la memoria. Tiene dos señales de control ya que se puede cargar desde dos sitios diferentes, desde el procesador o desde memoria.

→ Registro de direcciones (AR) Contiene la dirección que se carga cuando se va a leer o escribir en memoria.

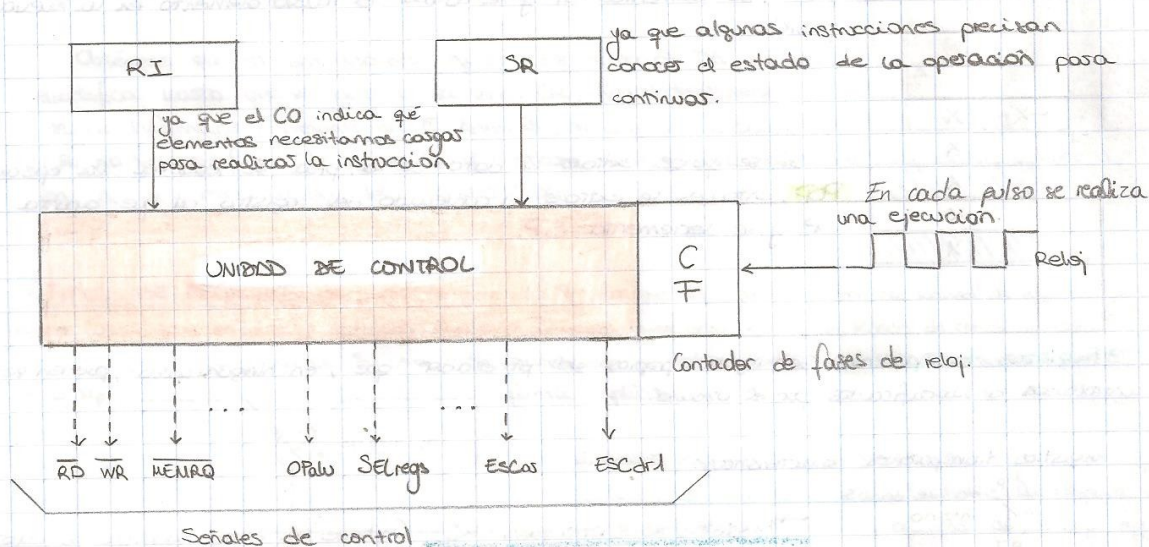
6

* CPU. UNIDAD DE CONTROL

Encargada de ir leyendo las instrucciones de la Mp, analizarlas (decodificarlas) y dar los órdenes a todos los componentes para realizar la ejecución. Genera todos los señales de los elementos de la máquina (señales de control).

Recibe las señales de los registros: SR (registro de estado) y RI (registro de instrucciones). Es un circuito síncrono.

La complejidad de la UC depende de la complejidad del juego de instrucciones de la máquina (nº de códigos diferentes, formatos de instrucciones (1 palabra, 1/2 palabras...) y modos de direccionamiento (formas distintas de describir la ubicación de los operandos)).

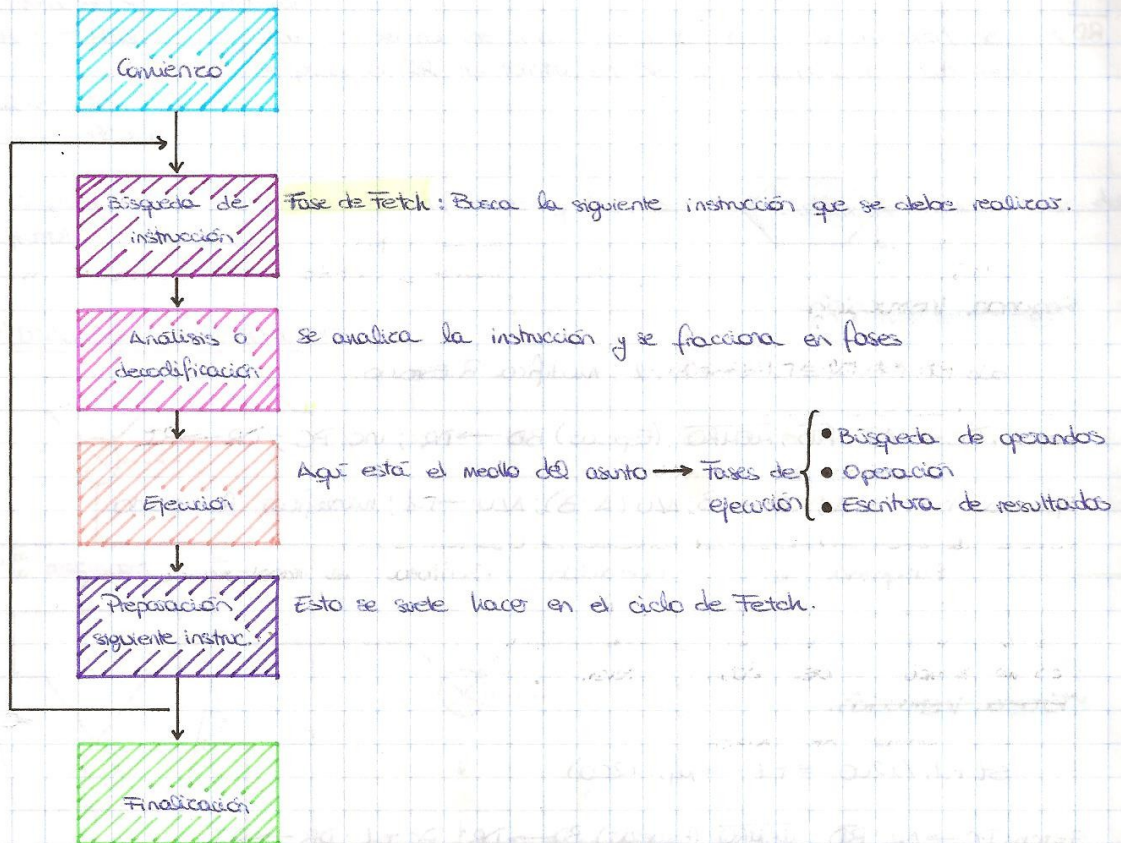


La UC indica en qué momento se activan las señales de control de cada instante. En su entrada se encuentra el registro de instrucciones (RI), el registro de estado (RS) y el reloj. En su salida se encuentran todas las señales de control del resto de componentes.

Cuando una instrucción vaya del registro de direcciones a la UC, se analiza y se activan las señales pertinentes. Las instrucciones se dividen en fases gracias al contador de fases. En cada fase se activan unas señales u otras.

Es importante tener un juego de instrucciones lo suficientemente flexible, simple y completo para poder hacer cualquier cosa sin saturar la UC y que trabaje lo más rápidamente posible.

1.2. FASES DE UNA EJECUCIÓN



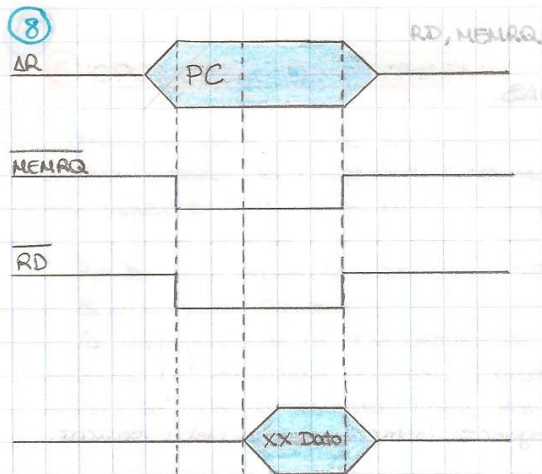
1ª Fase: La fase de Fetch es común a todas las instrucciones del mismo computador. La UC ordena la lectura de la Mp y carga la primera palabra (ó la única) en el IR e incrementa el PC. Dura varios ciclos de reloj.

2ª Fase: En la fase de análisis o decodificación la UC decodifica el CO y entonces se sabe que instrucción es. Dura un ciclo de CK.

3ª Fase: Por último, en la fase de ejecución, se hacen los operandos de la instrucción, se realizan las operaciones necesarias y se almacena el resultado. Esta fase es diferente para cada instrucción, incluso hay instrucciones que no tienen esta fase.

EJEMPLO 1.

Dirección	Ensamblador	primera instrucción
0	Load r1	$r1 \leftarrow Mp(1000)$
1	1000	Fetch $PC \rightarrow AR; RD, \overline{MEMRQ}; (Espeos) \rightarrow DR; DR \rightarrow AI;$
2	Subr1, r1, r2	incrementamos PC ($PC+1$)
3	Store r1	
4	1200	
5	Salta si Z	Siempre que se use la memoria y se lee en ella lo que se hace es similar, quitando la parte específica del ciclo de Fetch
6	50	
...	...	
1000	000...010	A continuación vendría la fase de análisis y decodificación.
...	...	Analisis: Se hace automáticamente en la UC
1200	000...101	Ejecución: lee 2ª palabra: $PC \rightarrow AR; RD, \overline{MEMRQ}; (Espeos) \rightarrow DR; DR \rightarrow AI; PC+1 \rightarrow \overline{PCNRQ}; \overline{RD} \rightarrow \overline{DR}; \overline{DR} \rightarrow \overline{AI}$



Segunda Instrucción

Sub $r1, r1, r2 \equiv r1 \leftarrow r1 - r2$; modifica R. Estado.

Fetch: $PC \rightarrow AR$; $\overline{RD}, \overline{MEMAQ}$; (Esperar) $BD \rightarrow DR$; inc PC; $DR \rightarrow RI$

Ejecución: $r1 \rightarrow A$; $r2 \rightarrow B$; $ALU(A-B)$; $ALU \rightarrow r1$; modifica R. Estado.

Búsqueda de operación Escritura de resultado
operandos

Tercera Instrucción

St $r1, 1200 \equiv r1 \rightarrow Mp(1200)$

Fetch: $PC \rightarrow AR$; $\overline{RD}, \overline{MEMAQ}$; (Esperar) $BD \rightarrow DR$; $PC+1$; $DR \rightarrow RI$

Ejecución: $PC \rightarrow AR$; $\overline{RD}, \overline{MEMAQ}$; (Esperar) $BD \rightarrow DR$; $DR \rightarrow AR$; $PC+1$; $r1 \rightarrow DR$; $\overline{RD}, \overline{MEMAQ}$; (Esperar)

Cuarta Instrucción

jmpe/50 \rightarrow Si $Z: 50 \rightarrow PC$; Si $\neg Z$ continúa.

Fetch: $PC \rightarrow AR$; $\overline{RD}, \overline{MEMAQ}$; (Esperar) $BD \rightarrow DR$; $PC+1$; Si $Z=1$; $DR \rightarrow PC$

EJEMPLO 2.

ADD .R1, .R2, .R3 $\equiv R1 \leftarrow R2 + R3$

010100...	0001	0010	0011
C0	R1	R2	R3

suprimimos direccionamiento de Mp a nivel de palabra.

Fetch: El procesador debe leer la instrucción de Mp. La dirección de dicha instrucción se encuentra en el PC.

$AR \leftarrow PC$

Lectura de Mp \overline{RD} , \overline{MEMAQ}

$DR \leftarrow M(AR)$

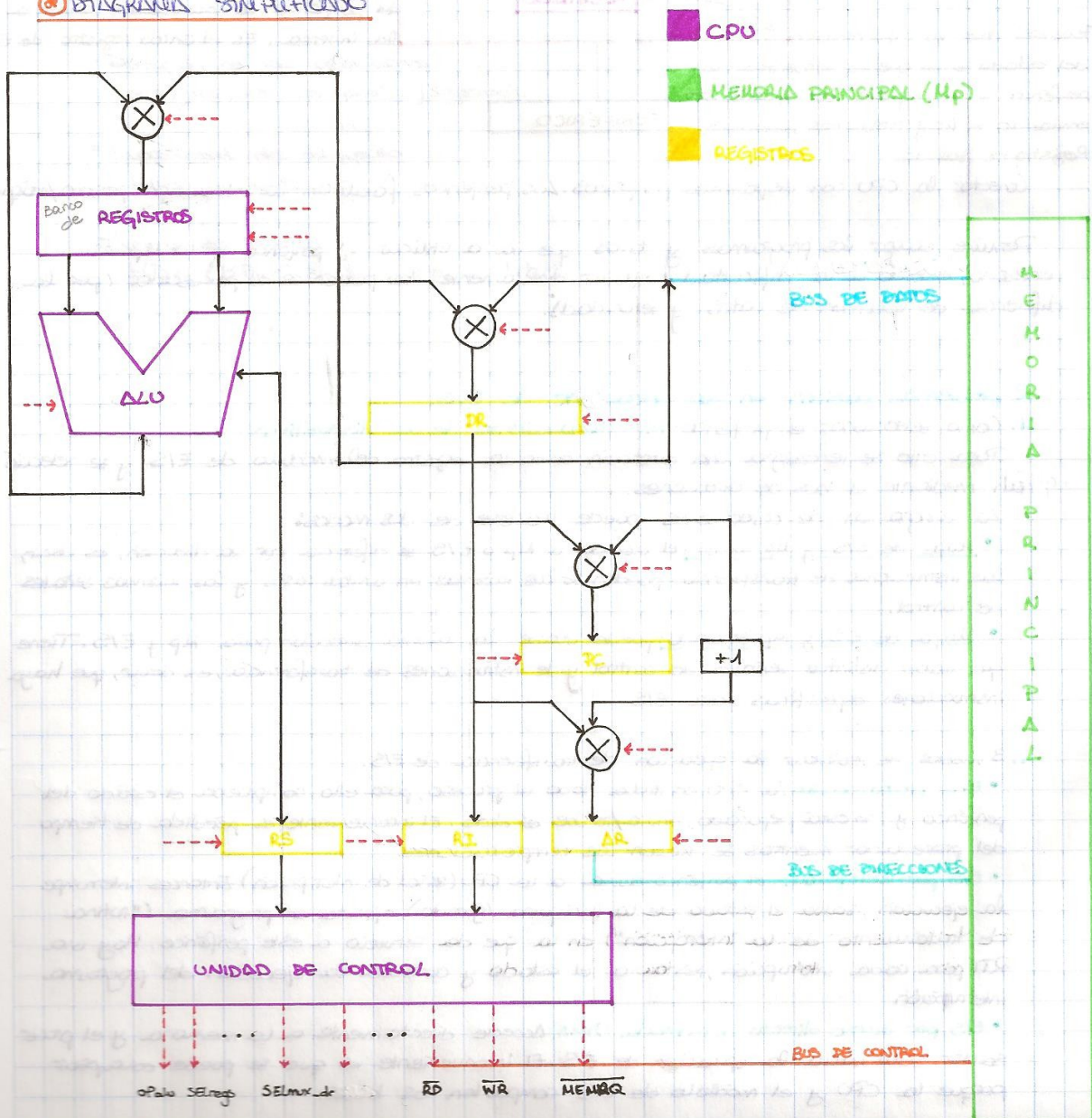
; Tras un tiempo de acceso, el contenido de la dirección de AR
; se pasa al DR, por tanto, en DR se encuentra la instrucción.

$IR \leftarrow DR$

$PC \leftarrow PC + 1$

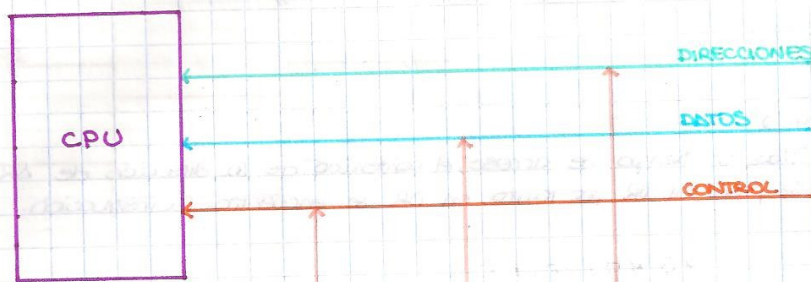
Después vendrá la fase de decodificación y la fase de ejecución que sumará los dos registros.

DIAGRAMA SIMPLIFICADO



10

1.3. UNIDAD DE ENTRADA / SALIDA



Envía los órdenes, parámetros e instrucciones de configuración al procesador. Es un registro de entrada.

Podemos leer la información del estado en el que se encuentra el periférico y el paquete de información enviando si hay problemas, velocidad, ... Registro de salida.

Mediante este registro se envían los datos, propiamente dichos, tanto del periférico al procesador como a la inversa. Es el único registro de E/S

Conecta la CPU con dispositivos periféricos. Los periféricos facilitan la interacción persona/máquina.

Permite cargar los programas y datos que va a utilizar el periférico en la Mp. Es necesario, debido a la dificultad de unir directamente los periféricos al procesador (por la diferencia de volumen de datos y velocidad).

2. Problemas básicos en las operaciones de E/S.

1. Cómo seleccionar el periférico: direccionamiento de los dispositivos

Para ello se le asigna una dirección a cada registro del módulo de E/S y se seleccionan mediante el bus de direcciones.

La asignación de direcciones puede hacerse de dos modos:

- Mapa de E/S y Mp común: el acceso a Mp o E/S se diferencia por la dirección, es decir, las instrucciones de transferencia pueden ser las mismas en ambos casos y las mismas señales de control.
- Mapa de E/S y Mp separado: puede usarse la misma dirección para Mp y E/S. Tiene que haber distintas señales de control y de instrucciones de transferencia, es decir, que haya instrucciones específicas para E/S.

2. 3 modos de realizar la operación de transferencia de E/S.

- E/S programada: la CPU controla todo el proceso, para ello comprueba el estado del periférico y, si está preparado, envía/recibe el dato. El inconveniente es pérdida de tiempo del procesador mientras se hacen las comprobaciones.
- E/S por interrupciones: El periférico avisa a la CPU (señal de interrupción). Entonces interrumpe la ejecución, salva el estado de la máquina (guarda), ejecuta un programa ("rutina de tratamiento de la interrupción") en la que da servicio a este periférico. Hay una RTI para cada interrupción, restaura el estado y continúa la ejecución del programa interrumpido.
- E/S por acceso directo a memoria (DMA): Accede directamente a la memoria y el procesador sólo inicia la operación de E/S. El inconveniente es que se puede colapsar porque la CPU y el módulo de E/S comparten los buses.

1.4. PARÁMETROS CARACTERÍSTICOS

- Byte (B): 1 Byte = 8 bits
- Ancho de palabra
 - 8, 16, 32, 64 bits (1B, 2B, 4B, 8B)
- Tamaño de la memoria
 - KB, MB, GB, TB (tera), PB (peta)
- Frecuencia del reloj.
 - MHz, GHz
- Duración de las operaciones
 - ms, μ s, ns, ps (pico), fs (femto)
- Capacidad de cómputo
 - MIPS, MFLOPS
- Ancho de banda
 - KB/s (KBps), MB/s, Kb/s, Mb/s